In this section, I'm going to give you another challenging exercise. So open up the Mocking folder, and look at this file. HouseKeeperHelper. In this class we have a public static method called SendStatementEmails.

The job of this method is to get a bunch of house keepers of the database. For each housekeeper it's going to generate a statement file and save it on the disc, and then it will eventually email that statement file to the housekeeper. So this is the big picture. Now if you look at the implementation of this class you can see these two methods, EmailFile and SaveStatement, they are both private methods.

As I told you before, look it's the same statement and you can see it's a private statement. So as I told you before, you should not test these private methods, they are implementation details. And here we have EmailFile again private static void. And the rest of the code in this file are just simple stubs I've generated to make this code compile. Because one of my students sent me the source code and I didn't have the entire code, so I had to reverse engineer it, and generate some classes to make the code compile. So don't worry about any of this stuff here. Your job is to write all the required unit tests for this method, SendStatementEmails. Now before you get started I want to give you a couple of things (?). These two methods, SaveStatement and EmailFile, they both touch external resources, so don't get stuck looking at the implementation of these methods. You need to extract these methods, put them in separate classes, and then use dependency injection to inject those dependencies inside this housekeeper helper class.

So, I want you to pause the video, and use what you have learned in this course to write all the required unit tests for this class. Next, we're going to look at my solution.